Error correcting code tester

Documentation

Short description of the application:

Client opens file given as an input parameter, encodes and sends it to server. Server introduces errors into received content (using method given by client) and sends it back to client. Client receives it and tries to restore correct data or informs that correction is not possible (statistics are provided).

How it works - step by step:

Server

1.Running server and verification of input data (port)

If user runs application without arguments, he has to type port number to connect. User also can choose port by typing it as a first argument in terminal/command line. If user gives more than one argument server return error.
Key in “--help” as argument to get help.

2.Creating socket and binding. Everything is automatically.

3. Server waits for a client to connect.

If any errors will occur, server return error.
If everything is correct and client connects to the server.

4.Getting initial parameters

Server waits for initial parameters (packet size, error generation method and additional information dependent of error generation method and receives it from client.

5.Getting packet from client, modifying it and resending.

When server gets package, modifies it on 1 of 3 ways and re sends to client.

1. Do nothing
2. Modifying every n bit
3. Modifying percentage of data

When everything is sent server waits for another client.

Client

1. Running client application and verification of input data

If user runs application without arguments, he has to type port number to connect and name of server host. User also can choose port by typing it as a first argument and host name as a second argument in terminal/command line.
If user gives more than two argument server returns error.

2.Creating socket and connecting to server (automatically)

3.Getting configurable parameters

Application asks user for configurable parameters (packet size, error generation method, error correcting method, name of the file to send) and verifies it. If everything went wrong server returns error.

4.Sending parameters to server

5.Encoding file or counting its CRC

Application uses chosen error correcting error and encodes file or count CRC.

6.Sending encoded packages and receiving modified by server.

Client sends encoded package, waits for modified packet from server in a loop, until it will send and receive all packets.

7.Decoding received packets or counting its CRC

Application decodes packets and saves received data to file called “output” in the main folder of Client.

8.Printing statistics

Program prints statistics of the transmission by comparing sent and received file and, if user asked, compares CRC’s of these files. It prints correctness in percentages.

9.Closing process